Exploiting sparsity in free energy basin-hopping

Kyle H. Sutherland-Cash[a], Rosemary G. Mantell[a], David J. Wales[a,*]

[a] *University Chemical Laboratories, Lensfield Road, Cambridge CB2 1EW, United Kingdom*

Abstract

We present acceleration of the vibrational frequency calculations required in free energy basin-hopping (FEBH) global optimisation using sparse Cholesky factorisation. We describe an interface with the SuiteSparse software package that facilitates these calculations, and present results detailing the accuracy and speedups obtained for atomic clusters and biomolecules. For the larger systems, exploiting sparsity reduces the computational cost by factors of ten to thirty, with no significant loss in accuracy. This approach will therefore provide access to the vibrational density of states, and hence to local harmonic free energies, for structure prediction and thermodynamic properties of more complex systems.

1. Introduction

Free energy basin-hopping (FEBH) [1] provides a means to locate the potential energy minimum corresponding to the lowest local free energy minimum of a molecular or condensed matter system at a specified temperature. The procedure is analogous to basin-hopping global optimisation [2, 3], but the accept/reject criterion is based upon the approximate free energy of the current minimum, rather than just the potential energy. Using the harmonic approximation within the superposition framework, the difference in free energy between the current (old) minimum and the new minimum is [1]

$$F_{\mathrm{new}}(T) - F_{\mathrm{old}}(T) = V_{\mathrm{new}} - V_{\mathrm{old}} + k_B T \ln \frac{o_{\mathrm{new}} \overline{\nu}^{\kappa}_{\mathrm{new}}}{o_{\mathrm{old}} \overline{\nu}^{\kappa}_{\mathrm{old}}}, \qquad (1)$$

*Corresponding author
Email address: dw34@cam.ac.uk (David J. Wales)

where F is the free energy, V is the potential energy, o is the point group order and $\overline{\nu}^{\kappa}$ is the product of vibrational normal mode frequencies. Hence, the mass-weighted Hessian and the log product of its non-zero eigenvalues must be calculated for each minimum in the chain of proposed moves. These additional calculations significantly increase the computational expense for each step involving a new minimum. In this contribution, we show how the calculation of the log product of vibrational normal modes of the Hessian matrix at a potential energy minimum can be accelerated by exploiting the sparsity of this matrix. Terms corresponding to rigid rotor degrees of freedom for molecular systems have a negligible effect for the systems considered here, so the vibrational density of states corresponding to this normal mode analysis is the key object of interest.

We routinely use the LAPACK DSYEV routine [4] for matrix diagonalisation and vibrational normal mode analysis, which provides the information required to calculate equilibrium thermodynamics and rates within the harmonic approximation [5]. However, the individual normal mode frequencies are not required in FEBH. Only the log product of positive Hessian eigenvalues (identical to the log of the determinant of the Hessian with zero eigenvalues shifted to unity) is needed within the harmonic approximation to the vibrational density of states. Hence, the DSYEV routine produces unnecessary information. Furthermore, the algorithm scales as $\mathcal{O}(N^3)$, making it relatively expensive for larger system sizes.

Georgescu and Mandelshtam [6] have tackled a similar problem, and they exploited two properties of the Hessian matrix to speed up their calculations. Firstly, the Hessian is symmetric positive definite if the zero eigenvalues corresponding to translation and rotation are excluded. This property means that a Cholesky factorisation can be used to uniquely decompose the matrix into a product of a lower triangular matrix and its conjugate transpose:

$$\mathbf{A} = \mathbf{L}\mathbf{L}^t. \tag{2}$$

The determinant of the Hessian can then be calculated from the product of

diagonals of $\mathbf{L}$ and $\mathbf{L}^t$. Secondly, the Hessian is expected to be sparse, with many negligible components, for reasonably short-ranged interatomic potentials. The two potentials we focus on here are the Lennard-Jones (LJ) form [7] and the AMBER force field [8]. The LJ pair potential is [7]

$$V(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right], \tag{3}$$

where ϵ is the well depth and $2^{1/6}\sigma$ is the pair equilibrium separation [9]. The second derivative tends to zero as r^{-8}, so components of the Hessian corresponding to interactions between distant particles are negligible.

The AMBER potential consists of bonded terms, which are short-ranged by definition, and non-bonded terms [10]. The van der Waals interactions are modelled using the LJ form and are hence relatively short-ranged. The electrostatic interactions are modelled as point charges and are longer-ranged. These interactions are often truncated through the use of a cutoff. However, the graphics processing unit (GPU) implementation of the AMBER 12 potential is used here [11, 12], which does not support cutoffs. For this reason, we take an alternative approach and simply set all values of the Hessian below a certain threshold to zero after the second derivatives have been calculated. Then, we employ a sparse Cholesky factorisation on CPU, instead of DSYEV, to find the log product of eigenvalues used in the FEBH calculations. Our results indicate that run times can be decreased by factors of ten to thirty for the larger examples considered, providing faster structure prediction, and access to more complex systems.

2. Methods

For the sparse implementation of FEBH, the usual call to DSYEV after the calculation of the mass-weighted Hessian was replaced with sparse routines. At a local minimum, the Hessian for an isolated molecule has six zero eigenvalues corresponding to overall translation and rotation, so the determinant will be zero. It is necessary to calculate the log product of the non-zero eigenvalues

corresponding to vibrational modes, so we apply a procedure that shifts the zero eigenvalues by $\lambda_{\text{shift}} = 1$. After ensuring that the Cartesian coordinates are in the centre of mass frame and coincident with the principal axes of rotation, we construct normalised eigenvectors, $\boldsymbol{\nu}$, corresponding to infinitesimal translations and rotations of the system (see AppendixA) [13]. We then shifted the corresponding Hessian eigenvalues according to the following equation:

$$H^{\dagger}_{\alpha\beta} = H_{\alpha\beta} + \lambda_{\text{shift}}\nu_{\alpha}\nu_{\beta}. \tag{4}$$

Before the Cholesky factorisation, all values of the Hessian below a specified magnitude were set to zero to ensure the sparsity of the matrix.

To perform the Cholesky factorisation, an interface to SuiteSparse (a package of sparse matrix solvers) was implemented [14]. The SuiteSparse library is mostly written in C, so an interface was constructed using the Fortran ISO C BINDING module to ensure interoperability between C and Fortran types, and to allow safe management of pointers and their associated memory. For our purposes, we required only functionality from the SuiteSparse CHOLMOD package [15]. The nonzero lower triangular elements of the Hessian were copied into a cholmod_triplet matrix, as triplets of nonzero values with their row and column indices. The `cholmod_triplet_to_sparse` function was used to convert this representation into a cholmod_sparse object, which was then passed to `cholmod_analyze`. This function selects a matrix ordering option that gives the best reduction in matrix fill-in and performs a symbolic factorisation. The most efficient algorithm to use for factorisation (either supernodal or simplicial) is automatically selected. The numeric factorisation was then performed using `cholmod_factorize`. The `cholmod_change_factor` function was then used to convert the returned factor to an $\mathbf{LDL}^t$ form, where $\mathbf{D}$ is diagonal. This procedure allowed the required diagonal entries to be accessed more easily, so that the determinant could then be calculated through accumulation of the logs of these values. The integration of SuiteSparse into our existing CMake build was greatly facilitated by the CMake scripts available on GitHub written by Jose Luis Blanco and Jerome Esnault [16].

3. Results and discussion

We have run tests to compare the calculation of the Hessian determinant using DSYEV and the new sparse Cholesky factorisation method as a function of system size for both LJ clusters and selected biomolecules represented by the AMBER force field [8]. Our objective was to achieve a significant speedup with minimal loss of accuracy. To this end, we wish to find the largest magnitude for the Hessian cutoff that still provides sufficient accuracy. If the Hessian cutoff is too large, not only do we lose accuracy, but the Cholesky factorisation can fail outright because the matrix is no longer positive definite. Factorisation failures due to non-positive definite matrices can also occur if the minima are not converged tightly enough, so the root mean square (RMS) force convergence criterion for the local minima is also an important parameter. For a given set of parameters, even if the calculations are still reasonably accurate, too many factorisation failures will reduce the efficiency of the FEBH procedure. We regard a proportion of factorisation failures below 1% as ideal.

The GMIN FEBH code [17] was compiled on the x86_64 architecture running the Ubuntu 14.04.4 operating system, using Intel compiler version 16.0.4 with the MKL library version 11.3.4. The tests were run on 2.40 GHz Intel Xeon E5-2620 v3 CPUs and employed version 4.5.3 of SuiteSparse.

3.1. Atomic clusters described by the Lennard-Jones potential

Tables 1a and 1b show the percentage of factorisation failures for LJ_{1000} and LJ_{5000} clusters respectively, for various combinations of the RMS force convergence criterion for local minimisation and the Hessian cutoff. These percentages are calculated from 100 FEBH runs of nine steps each, starting from randomly generated atomic configurations, giving 1000 local minima. The FEBH step size was set to a maximum of $0.43\,\sigma$ for any individual Cartesian coordinate, and angular steps were taken for individual LJ atoms above a pair binding energy tolerance of $0.52\,\epsilon$. For both cluster sizes, the values for the RMS force were chosen to span a range showing the effects of both 'under-convergence'

and 'over-convergence'. The effect of an RMS force that is too large can be seen in the 10^{-3} row of Table 1a and the 10^{-4} row of Table 1b, both of which exhibit factorisation failure frequencies greater than 1%. Similarly, the Hessian cutoff spans a range of values that demonstrate the effects of a cutoff that is both too large and too small. An increased number of factorisation failures is observed for a Hessian cutoff value that is too large, such as in the 10^{-2} column of Table 1a and the 2×10^{-3} column of Table 1b. A visual representation of the effect of varying the Hessian cutoff on overall sparsity for LJ_{1000} is shown in Figure 1.

(a) LJ_{1000}

RMS force / $\epsilon\sigma^{-1}$	**Hessian cutoff** / $\epsilon\sigma^{-2}$		
	10^{-6}	10^{-3}	10^{-2}
10^{-3}	4.1	4.1	18.2
10^{-5}	0.7	0.7	15.3
10^{-7}	0.9	0.9	13.3

(b) LJ_{5000}

RMS force / $\epsilon\sigma^{-1}$	**Hessian cutoff** / $\epsilon\sigma^{-2}$		
	10^{-6}	10^{-3}	2×10^{-3}
10^{-4}	3.2	3.1	11.4
10^{-6}	0.4	0.4	6.5
10^{-8}	0.0	0.0	4.3

Table 1: Percentage of factorisation failures (resulting from a non-positive definite Hessian) for two LJ cluster sizes as a function of RMS force convergence threshold and the cutoff for neglect of Hessian matrix elements.

We aim for the calculation of the log product of Hessian eigenvalues using the sparse matrix method to produce essentially the same result as full diagonalisation. The percentage errors in the sparse calculations relative to diagonalisation were found to be much less than 1% for all combinations of parameters. This high level of accuracy can be accounted for by the fact that the magnitude of the neglected Hessian matrix elements is indeed very close to zero. These values are many orders of magnitude smaller than the largest values in the matrix, and so contribute little to the final result.

The principal objective of this study was to speed up the calculation of the log product of eigenvalues of the Hessian, and Tables 2a and 2b summarise the results. The values are reported as the average time for a DSYEV calculation divided by the average time for a sparse calculation from the same basin-hopping runs used to produce all the preceding results. Timings for the calculations that failed were not included in the averages. As expected, the speedups are greater for larger values of the Hessian cutoff, due to the greater sparsity of the resulting matrices. The RMS force threshold does not seem to affect the speedups obtained for either system to a large extent. Greater efficiency gains were obtained for the large Hessian matrices associated with the larger LJ_{5000} cluster.

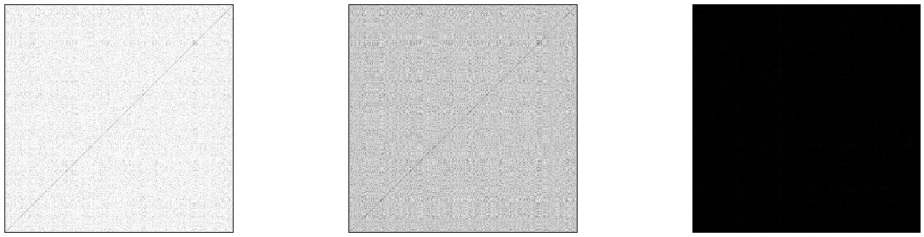
(a) LJ_{1000}

RMS force / $\epsilon\sigma^{-1}$	**Hessian cutoff** / $\epsilon\sigma^{-2}$		
	10^{-6}	10^{-3}	10^{-2}
10^{-3}	2.9	4.6	7.7
10^{-5}	2.8	4.5	7.8
10^{-7}	2.8	4.5	7.7

(b) LJ_{5000}

RMS force / $\epsilon\sigma^{-1}$	**Hessian cutoff** / $\epsilon\sigma^{-2}$		
	10^{-6}	10^{-3}	2×10^{-3}
10^{-4}	7.7	34.0	37.7
10^{-6}	9.2	35.2	36.8
10^{-8}	10.1	36.2	38.1

Table 2: Average speedups for finding the log product of positive eigenvalues for two LJ cluster sizes as a function of RMS force convergence threshold and the cutoff for neglect of Hessian matrix elements.



(a) $10^{-2}\,\epsilon\sigma^{-2}$ (b) $10^{-3}\,\epsilon\sigma^{-2}$ (c) $10^{-6}\,\epsilon\sigma^{-2}$

Figure 1: Visualisation of cutoff variation for neglect of Hessian matrix elements on the sparsity of the resulting matrix. The plots are derived from a single LJ_{1000} minimum converged to an RMS force tolerance of $10^{-5}\,\epsilon\sigma^{-1}$. Non-zero matrix elements are coloured black and matrix elements set to zero are coloured white. Each subfigure is labelled with the appropriate value for the cutoff.

3.2. Test cases for proteins

The activity of biological systems is strongly temperature dependent, so these molecules constitute an important application for structure prediction using FEBH [1]. The systems investigated here are a truncated monomeric version (3522 atoms) of the trimeric haemagglutinin (HA) glycoprotein of the influenza A(H1N1) virus and a full monomeric structure (7585 atoms) [18]. The symmetrised [19, 20] AMBER ff99SB forcefield [8] was used in both cases with an effectively infinite non-bonded cutoff (999.99 Å). All tests used the modified GB solvent model [21, 22] (AMBER input flag igb = 2) at a salt concentration 0.2 M with a cutoff of 12 Å for the calculation of the effective Born radii. The tests considered here employ 100 starting structures each, taken from high temperature molecular dynamics (MD) runs. For the truncated monomer, nine basin-hopping steps were taken for each structure to generate 1000 local minima, with Cartesian moves of the backbone atoms set to a maximum value of 0.2 Å. However, due to the computational expense of the DSYEV calculations for the full monomer, no basin-hopping steps were performed for these starting structures and only 100 minimised structures were obtained. To reduce the overall time taken for the tests, the minimisations were carried out using a GPU-

accelerated implementation [23] of L-BFGS [24] (the limited-memory version of the BFGS algorithm, named for Broyden [25], Fletcher [26], Goldfarb [27] and Shanno [28]). GeForce GTX TITAN Black GPUs were used, with NVIDIA Linux driver version 375.26 and NVIDIA CUDA Toolkit 8.0.

Due to the lack of GPU-accelerated analytical second derivatives for the AMBER potential, numerical second derivatives with calls to the GPU-accelerated AMBER energy and gradient subroutine were used in calculating the Hessian matrix. A central difference approximation of the form

$$\frac{\partial^2 V}{\partial x_i x_j} \approx \frac{\nabla_i V(\mathbf{X} + \zeta \mathbf{x}) - \nabla_i V(\mathbf{X} - \zeta \mathbf{x})}{2\zeta x_j} \tag{5}$$

was employed in calculating these derivatives, where $V(\mathbf{X})$ is the energy at configuration $\mathbf{X}$ in nuclear configuration space and $\zeta \ll 1$. We performed some preliminary tests to determine the optimal value of ζ for the final benchmarks. We minimised the 100 full monomeric structures for a range of values of ζ and recorded the percentage of factorisation failures resulting from a non-positive definite Hessian matrix. Based on the results in Table 3, we chose a value of 10^{-4} for ζ.

Table 3: Percentage of factorisation failures (resulting from a non-positive definite Hessian) for the HA full monomer as a function of the value of ζ used in calculating the numerical second derivatives. A value of 10^{-6} kcal mol^{-1} Å^{-1} was used for the RMS force convergence threshold and the cutoff for neglect of Hessian matrix elements was set to 10^{-6} kcal mol^{-1} Å^{-2}.

ζ	**% of factorisation failures**
10^{-2}	5
10^{-3}	3
10^{-4}	0
10^{-5}	11
10^{-6}	25
10^{-7}	9

We also tested the SuiteSparse GPU implementation [29] to accelerate our calculations. However, we opted not to use this approach in our tests as the speedup only amounted to about 5–10% for the examples considered here. This result is likely due to the relatively small size of our matrices compared to those benchmarked by the developers, for both the clusters and proteins considered here.

Tables 4a and 4b show the percentage of factorisation failures for the truncated monomer and the full monomer with various values of the RMS force convergence criterion and the Hessian cutoff. As for the atomic clusters considered in Section 3.1, fewer factorisation failures are observed for smaller values of the RMS force and Hessian cutoff.

(a) Truncated HA monomer

RMS force / kcal mol^{-1} Å^{-1}	**Hessian cutoff / kcal mol^{-1} Å^{-2}**		
	10^{-8}	10^{-5}	3.5×10^{-5}
10^{-4}	1.8	1.8	6.3
10^{-5}	0.8	0.8	5.2
10^{-6}	0.0	0.0	4.7

(b) Full HA monomer

RMS force / kcal mol^{-1} Å^{-1}	**Hessian cutoff / kcal mol^{-1} Å^{-2}**		
	10^{-9}	10^{-6}	6.0×10^{-6}
10^{-4}	3.0	3.0	24.0
10^{-5}	0.0	0.0	20.0
10^{-6}	0.0	0.0	19.0

Table 4: Percentage of factorisation failures (resulting from a non-positive definite Hessian) for two protein examples as a function of RMS force convergence threshold and the cutoff for neglect of Hessian matrix elements.

The calculations of the log product of eigenvalues were highly accurate for all combinations of parameters, with average percentage errors below 1%. The associated speedups are shown in Tables 5a and 5b. The efficiency gains with respect to Hessian cutoff and RMS force are similar to the results we obtained for the LJ clusters.

(a) Truncated HA monomer

RMS force / kcal mol^{-1} Å^{-1}	**Hessian cutoff / kcal mol^{-1} Å^{-2}**		
	10^{-8}	10^{-5}	3.5×10^{-5}
10^{-4}	7.3	7.7	8.4
10^{-5}	7.1	7.8	8.6
10^{-6}	7.3	7.7	8.5

(b) Full HA monomer

RMS force / kcal mol^{-1} Å^{-1}	**Hessian cutoff / kcal mol^{-1} Å^{-2}**		
	10^{-9}	10^{-6}	6.0×10^{6}
10^{-4}	14.6	16.9	17.3
10^{-5}	14.6	16.8	16.6
10^{-6}	14.8	16.7	18.0

Table 5: Average speedups for finding log product of eigenvalues for two protein examples as a function of RMS force convergence threshold and the cutoff for neglect of Hessian matrix elements.

4. Conclusions

This contribution has described the implementation of a new interface between our free energy basin-hopping (FEBH) global optimisation routines and the SuiteSparse package, which allows us to perform sparse Cholesky factorisation to obtain vibrational frequencies within the FEBH framework. Tests were performed for both LJ clusters and selected biomolecules represented using the AMBER potential. A significant speedup was achieved for the calculation of the vibrational density of states from the product of non-zero Hessian eigenvalues, with negligible loss of accuracy, with respect to the fastest matrix diagonalisation routine that does not exploit sparsity. Exploiting sparsity will therefore accelerate structure prediction and provide access to larger and more complex systems. This capability may be important for a wide range of molecular and condensed matter systems where finite temperature effects are significant. It enables us to estimate local free energies, along with other potentially useful thermodynamic properties, such as the heat capacity. We are currently exploiting these results to investigate the effect of mutations on the properties of proteins and nucleic acids. Another possibility for future work would be to apply this method to systems with longer-range interactions. This would enable us to compare the range of validity of these sparse methods in important systems such as metallic and ionic clusters.

Notes

Additional data related to this publication is available on Zenodo: https://doi.org/10.5281/zenodo.814244

Acknowledgements

We gratefully acknowledge funding from the EPSRC (grant references EP/L504920/1 and EP/1001352/1).

AppendixA. Shifting the Hessian eigenvalues

The translational and rotational eigenvalues must be shifted from zero to one at each local minimum of interest to obtain the product of positive Hessian eigenvalues. To do this, we must find normal mode vectors for translation and rotation, where the molecule is aligned so that the principal axes and the centre of mass cooincide with the fixed axis system and the origin. The corresponding Hessian eigenvectors are then orthogonal, as detailed below.

Let $X_{\alpha x}$ be the x coordinate of atom α, etc. A rotation of the position vector of atom α, $\mathbf{X}^0_\alpha$, through angle θ about an axis defined by the unit vector $\hat{\mathbf{n}}$ is given by

$$\mathbf{X}_\alpha = \mathbf{X}^0_\alpha \cos\theta + \hat{\mathbf{n}}(\hat{\mathbf{n}} \cdot \mathbf{X}^0_\alpha)(1 - \cos\theta) + \mathbf{X}^0_\alpha \wedge \hat{\mathbf{n}} \sin\theta. \tag{A.1}$$

Taking a Taylor series about $\theta = 0$, the displacement vector for this rotation is

$$\boldsymbol{\Delta}_\alpha = \mathbf{X}_\alpha - \mathbf{X}^0_\alpha = \theta \begin{pmatrix} \hat{n}_z X^0_{\alpha y} - \hat{n}_y X^0_{\alpha z} \\ \hat{n}_x X^0_{\alpha z} - \hat{n}_z X^0_{\alpha x} \\ \hat{n}_y X^0_{\alpha x} - \hat{n}_x X^0_{\alpha y} \end{pmatrix} + \frac{1}{2}\theta^2 \begin{pmatrix} -X^0_{\alpha x} + \hat{n}_x^2 X^0_{\alpha x} + \hat{n}_x(\hat{n}_y X^0_{\alpha y} + \hat{n}_z X^0_{\alpha z}) \\ -X^0_{\alpha y} + \hat{n}_y^2 X^0_{\alpha y} + \hat{n}_y(\hat{n}_x X^0_{\alpha x} + \hat{n}_z X^0_{\alpha z}) \\ -X^0_{\alpha z} + \hat{n}_z^2 X^0_{\alpha z} + \hat{n}_z(\hat{n}_x X^0_{\alpha x} + \hat{n}_y X^0_{\alpha y}) \end{pmatrix} \tag{A.2}$$

A Taylor expansion of the potential energy gives

$$V(\mathbf{X}^0) = V(\mathbf{X}^0 + \boldsymbol{\Delta}) = V(\mathbf{X}^0) + \mathbf{G}(\mathbf{X}^0)^T \boldsymbol{\Delta} + \frac{1}{2}\boldsymbol{\Delta}^T \mathbf{H}(\mathbf{X}^0)\boldsymbol{\Delta} + \ldots, \tag{A.3}$$

where $\boldsymbol{\Delta} = \{\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2, \ldots, \boldsymbol{\Delta}_N\}^T$, $\mathbf{G}(\mathbf{X}^0) = \nabla V(\mathbf{X}^0)$, or $G_\alpha(\mathbf{X}^0) = \partial V(\mathbf{X}^0)/\partial X_\alpha$, and $H_{\alpha\beta}(\mathbf{X}^0) = \partial^2 V(\mathbf{X}^0)/\partial X_\alpha \partial X_\beta$. The potential energy of an isolated molecule must be invariant to rotation, so the terms in $\boldsymbol{\Delta}$ must be equal to zero. At a stationary point, $\mathbf{G}(\mathbf{X}^0) = \mathbf{0}$ and equating terms in α^2 to zero gives

$$\mathbf{H}(\mathbf{X}^0)\boldsymbol{\Delta} = \begin{pmatrix} \hat{n}_z X^0_{1y} - \hat{n}_y X^0_{1z} \\ \hat{n}_x X^0_{1z} - \hat{n}_z X^0_{1x} \\ \hat{n}_y X^0_{1x} - \hat{n}_x X^0_{1y} \\ \vdots \end{pmatrix} = \mathbf{0}. \tag{A.4}$$

This result implies that $\boldsymbol{\Delta}$ is an eigenvector of $\mathbf{H}$ with eigenvalue zero. We require eigenvectors of the mass-weighted Hessian corresponding to mass-weighted

coordinates, $\mathbf{Q}_X = \mathbf{X}_\alpha\sqrt{m_\alpha}$, where m_α is the mass of atom α, and the corresponding components are

$$\begin{pmatrix} \sqrt{m_1}(\hat{n}_z X^0_{1y} - \hat{n}_y X^0_{1z}) \\ \sqrt{m_1}(\hat{n}_x X^0_{1z} - \hat{n}_z X^0_{1x}) \\ \sqrt{m_1}(\hat{n}_y X^0_{1x} - \hat{n}_x X^0_{1y}) \\ \vdots \end{pmatrix}. \tag{A.5}$$

We can prove that the rotational eigenvectors are orthogonal if we choose rotations about directions corresponding to eigenvectors of the moment of inertia tensor:

$$\mathbf{I} = \sum_\alpha m_\alpha \begin{pmatrix} (X^0_{\alpha y})^2 + (X^0_{\alpha z})^2 & -X^0_{\alpha x}X^0_{\alpha y} & -X^0_{\alpha x}X^0_{\alpha z} \\ -X^0_{\alpha y}X^0_{\alpha x} & (X^0_{\alpha x})^2 + (X^0_{\alpha z})^2 & -X^0_{\alpha y}X^0_{\alpha z} \\ -X^0_{\alpha z}X^0_{\alpha x} & -X^0_{\alpha z}X^0_{\alpha y} & (X^0_{\alpha x})^2 + (X^0_{\alpha y})^2 \end{pmatrix}. \tag{A.6}$$

Using equation (A.5) with two eigenvectors of $\mathbf{I}$ corresponding to $\mathbf{I}\hat{\mathbf{e}}_a = \lambda_a\hat{\mathbf{e}}_a$ and $\mathbf{I}\hat{\mathbf{e}}_b = \lambda_b\hat{\mathbf{e}}_b$, replacing $\hat{\mathbf{n}}$, we can construct the dot product as:

$$\begin{aligned} & \sum_\alpha m_\alpha \big[(\hat{e}_{az}X^0_{\alpha y} - \hat{e}_{ay}X^0_{\alpha z})(\hat{e}_{bz}X^0_{\alpha y} - \hat{e}_{by}X^0_{\alpha z}) + (\hat{e}_{ax}X^0_{\alpha z} - \hat{e}_{az}X^0_{\alpha x})(\hat{e}_{bx}X^0_{\alpha z} - \hat{e}_{bz}X^0_{\alpha x}) \\ & \qquad +(\hat{e}_{ay}X^0_{\alpha x} - \hat{e}_{ax}X^0_{\alpha y})(\hat{e}_{by}X^0_{\alpha x} - \hat{e}_{bx}X^0_{\alpha y})\big] \\ = & \sum_\alpha m_\alpha \big[\hat{e}_{ax}\hat{e}_{bx}\left\{(X^0_{\alpha z})^2 + (X^0_{\alpha y})^2\right\} - \hat{e}_{by}\hat{e}_{ax}X^0_{\alpha x}X^0_{\alpha y} - \hat{e}_{bz}\hat{e}_{ax}X^0_{\alpha x}X^0_{\alpha z} \\ & \qquad +\hat{e}_{ay}\hat{e}_{by}\left\{(X^0_{\alpha z})^2 + (X^0_{\alpha x})^2\right\} - \hat{e}_{bz}\hat{e}_{ay}X^0_{\alpha y}X^0_{\alpha z} - \hat{e}_{bx}\hat{e}_{ay}X^0_{\alpha y}X^0_{\alpha x} \\ & \qquad +\hat{e}_{az}\hat{e}_{bz}\left\{(X^0_{\alpha x})^2 + (X^0_{\alpha y})^2\right\} - \hat{e}_{bx}\hat{e}_{az}X^0_{\alpha z}X^0_{\alpha x} - \hat{e}_{by}\hat{e}_{az}X^0_{\alpha z}X^0_{\alpha y}\big] \\ = & \hat{\mathbf{e}}_a^T\mathbf{I}\hat{\mathbf{e}}_b = \hat{\mathbf{e}}_a^T\lambda_b\hat{\mathbf{e}}_b = \lambda_b\hat{\mathbf{e}}_a^T\hat{\mathbf{e}}_b = 0. \end{aligned} \tag{A.7}$$

The dot product of an infinitesimal rotation and a translation corresponding to axes defined by eigenvectors $\hat{\mathbf{e}}_a$ and $\hat{\mathbf{e}}_b$ of $\mathbf{I}$ is

$$\begin{aligned} & \sum_\alpha m_\alpha[(\hat{e}_{az}X^0_{\alpha y} - \hat{e}_{ay}X^0_{\alpha z})\hat{e}_{bx} + (\hat{e}_{ax}X^0_{\alpha z} - \hat{e}_{az}X^0_{\alpha x})\hat{e}_{by} + (\hat{e}_{ay}X^0_{\alpha x} - \hat{e}_{ax}X^0_{\alpha y})\hat{e}_{bz}] \\ = & (\hat{e}_{ay}\hat{e}_{bz} - \hat{e}_{az}\hat{e}_{by})\sum_\alpha m_\alpha X^0_{\alpha x} + (\hat{e}_{az}\hat{e}_{bx} - \hat{e}_{ax}\hat{e}_{bz})\sum_\alpha m_\alpha X^0_{\alpha y} + (\hat{e}_{ax}\hat{e}_{by} - \hat{e}_{ay}\hat{e}_{bx})\sum_\alpha m_\alpha X^0_{\alpha z} \\ = & (\hat{e}_a \wedge \hat{e}_b)\cdot\sum_\alpha m_\alpha\mathbf{X}^0 = 0, \end{aligned} \tag{A.8}$$

when the centre-of-mass is at the origin, because $\sum_\alpha m_\alpha \mathbf{X}^0 = 0$. Hence the Hessian eigenvectors corresponding to infinitesimal rotations and translations are orthogonal if the system centre of mass is at the origin and the system is oriented so that the principal axes of the inertia tensor are aligned with the global frame. This alignment enables us to shift the six corresponding eigenvalues to unity individually.

References

[1] K. H. Sutherland-Cash, D. J. Wales, D. Chakrabarti, Free energy basin-hopping, Chem. Phys. Lett. 625 (2015) 1–4. doi:10.1016/j.cplett.2015.02.015.

[2] Z. Li, H. A. Scheraga, Monte Carlo-minimization approach to the multiple-minima problem in protein folding, Proc. Natl. Acad. Sci. U.S.A. 84 (19) (1987) 6611–6615.

[3] D. J. Wales, J. P. K. Doye, Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms, J. Phys. Chem. A 101 (28) (1997) 5111–5116. doi:10.1021/jp970984n.

[4] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, LAPACK Users' Guide, 3rd Edition, Society for Industrial and Applied Mathematics, Philadelphia, 1999.

[5] D. J. Wales, Energy Landscapes, Cambridge University Press, Cambridge, 2003.

[6] I. Georgescu, V. A. Mandelshtam, Self-consistent phonons revisited. I. The role of thermal versus quantum fluctuations on structural transitions in large Lennard-Jones clusters, J. Chem. Phys. 137 (14) (2012) 144106. doi:10.1063/1.4754819.

[7] F. M. Mourits, F. H. A. Rummens, A critical evaluation of Lennard-Jones and Stockmayer potential parameters and of some correlation methods, Can. J. Chem. 55 (16) (1977) 3007–3020. doi:10.1139/v77-418.

[8] V. Hornak, R. Abel, A. Okur, B. Strockbine, A. Roitberg, C. Simmerling, Comparison of Multiple Amber Force Fields and Development of Improved Protein Backbone Parameters, Proteins: Struct., Funct., Bioinf. 65 (3) (2006) 712–725. doi:10.1002/prot.21123.

[9] J. E. Jones, On the Determination of Molecular Fields. II. From the Equation of State of a Gas, Proc. R. Soc. Lond. A 106 (738) (1924) 463–477. doi:10.1098/rspa.1924.0082.

[10] D. A. Pearlman, D. A. Case, J. W. Caldwell, W. S. Ross, T. E. Cheatham III, S. DeBolt, D. Ferguson, G. Seibel, P. Kollman, AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules, Comput. Phys. Commun. 91 (1) (1995) 1–41. doi:http://dx.doi.org/10.1016/0010-4655(95)00041-D.

[11] D. A. Case, T. A. Darden, T. E. Cheatham III, C. L. Simmerling, J. Wang, R. E. Duke, R. Luo, R. C. Walker, W. Zhang, K. M. Merz, B. Roberts, S. Hayik, A. Roitberg, G. Seabra, J. Swails, A. W. Goetz, I. Kolossvai, K. F. Wong, F. Paesani, J. Vanicek, R. M. Wolf, J. Liu, X. Wu, S. R. Brozell, T. Steinbrecher, H. Gohlke, Q. Cai, X. Ye, J. Wang, M. J. Hsieh, G. Cui, D. R. Roe, D. H. Mathews, M. G. Seetin, R. Salomon-Ferrer, C. Sagui, V. Babin, T. Luchko, S. Gusarov, A. Kovalenko, P. A. Kollman, AMBER 12, University of California, San Francisco, 2012.

[12] A. W. Götz, M. J. Williamson, D. Xu, D. Poole, S. Le Grand, R. C. Walker, Routine Microsecond Molecular Dynamics Simulations with AMBER on GPUs. 1. Generalized Born, J. Chem. Theory Comput. 8 (5) (2012) 1542–1555. doi:10.1021/ct200909j.

[13] M. Page, J. W. McIver, On evaluating the reaction path Hamiltonian, J. Chem. Phys. 88 (2) (1988) 922–935. doi:10.1063/1.454172.

[14] SuiteSparse, http://faculty.cse.tamu.edu/davis/suitesparse.html (accessed Jun 18, 2017).

[15] Y. Chen, T. A. Davis, W. W. Hager, S. Rajamanickam, Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate, ACM Trans. Math. Softw. 35 (3) (2008) 22. doi:10.1145/1391989.1391995.

[16] suitesparse-metis-for-windows, https://github.com/jlblancoc/suitesparse-metis-for-windows (accessed Jun 18, 2017).

[17] GMIN: A program for finding global minima and calculating thermodynamic properties from basin-sampling., http://www-wales.ch.cam.ac.uk/GMIN/ (accessed Jun 18, 2017).

[18] S. Chutinimitkul, S. Herfst, J. Steel, A. C. Lowen, J. Ye, D. v. Riel, E. J. A. Schrauwen, T. M. Bestebroer, B. Koel, D. F. Burke, K. H. Sutherland-Cash, C. S. Whittleston, C. A. Russell, D. J. Wales, D. J. Smith, M. Jonges, A. Meijer, M. Koopmans, G. F. Rimmelzwaan, T. Kuiken, A. D. M. E. Osterhaus, A. García-Sastre, D. R. Perez, R. A. M. Fouchier, Virulence-Associated Substitution D222G in the Hemagglutinin of 2009 Pandemic Influenza A(H1N1) Virus Affects Receptor Binding, J. Virol. 84 (22) (2010) 11802–11813. doi:10.1128/JVI.01136-10.

[19] E. Małolepsza, B. Strodel, M. Khalili, S. Trygubenko, S. N. Fejer, D. J. Wales, Symmetrization of the AMBER and CHARMM force fields, J. Comput. Chem. 31 (7) (2010) 1402–1409. doi:10.1002/jcc.21425.

[20] E. Małolepsza, B. Strodel, M. Khalili, S. Trygubenko, S. Fejer, J. M. Carr, D. J. Wales, Erratum: Symmetrization of the AMBER and CHARMM force fields [J. Comp. Chem. 31, 1402], J. Comput. Chem. 33 (27) (2012) 2209–2209. doi:10.1002/jcc.23064.

[21] A. Onufriev, D. Bashford, D. A. Case, Modification of the Generalized Born Model Suitable for Macromolecules, J. Phys. Chem. B 104 (15) (2000) 3712–3720. doi:10.1021/jp994072s.

[22] A. Onufriev, D. Bashford, D. A. Case, Exploring protein native states and large-scale conformational changes with a modified generalized born model, Proteins: Struct., Funct., Bioinf. 55 (2) (2004) 383–394. doi:10.1002/prot.20033.

[23] R. G. Mantell, C. E. Pitt, D. J. Wales, GPU-Accelerated Exploration of Biomolecular Energy Landscapes, J. Chem. Theory Comput. 12 (12) (2016) 6182–6191. doi:10.1021/acs.jctc.6b00934.

[24] J. Nocedal, Updating quasi-Newton matrices with limited storage, Math. Comput. 35 (151) (1980) 773–782. doi:10.1090/S0025-5718-1980-0572855-7.

[25] C. G. Broyden, The Convergence of a Class of Double-rank Minimization Algorithms: 2. The New Algorithm, IMA J. Appl. Math. 6 (3) (1970) 222–231. doi:10.1093/imamat/6.3.222.

[26] R. Fletcher, A new approach to variable metric algorithms, Comput. J. 13 (3) (1970) 317–322. doi:10.1093/comjnl/13.3.317.

[27] D. Goldfarb, A Family of Variable-Metric Methods Derived by Variational Means, Math. Comput. 24 (109) (1970) 23–26. doi:10.2307/2004873.

[28] D. F. Shanno, Conditioning of Quasi-Newton Methods for Function Minimization, Math. Comput. 24 (111) (1970) 647–656. doi:10.2307/2004840.

[29] S. C. Rennich, D. Stosic, T. A. Davis, Accelerating sparse Cholesky factorization on GPUs, Parallel Comput. 59 (2016) 140–150. doi:10.1016/j.parco.2016.06.004.